Int. J. Appl. Math. Comput. Sci., 2011, Vol. 21, No. 2, 295–306
DOI: 10.2478/v10006-011-0022-1

P2P VIDEO STREAMING COMBINING SVC AND MDC

FRANCISCO DE ASÍS LÓPEZ-FUENTES

Department of Information Technology, Autonomous Metropolitan University, Cuajimalpa
Av. Constituyentes 1054, Col. Lomas Altas, 11950, Mexico, D.F., Mexico
e-mail: flopez@correo.cua.uam.mx

In this paper we propose and evaluate a combined SVC-MDC (Scalable Video Coding & Multiple Description Video Coding) video coding scheme for Peer-to-Peer (P2P) video multicast. The proposed scheme is based on a full cooperation established between the peer sites, which contribute their upload capacity during video distribution. The source site splits the video content into many small blocks and assigns each block to a single peer for redistribution. Our solution is implemented in a fully meshed P2P network in which peers are connected to each other via UDP (User Datagram Protocol) links. The video content is encoded by using the Scalable Video Coding (SVC) method. We present a flow control mechanism that allows us to optimize dynamically the overall throughput and to automatically adjust video quality for each peer. Thus, peers with different upload capacity receive different video quality. We also combine the SVC method with Multiple Description Coding (MDC) to alleviate the packet loss problem. We implemented and tested this approach in the PlanetLab infrastructure. The obtained results show that our solution achieves good performance and remarkable video quality in the presence of packet loss.

Keywords: peer-to-peer networks, video streaming, scalable video coding, multiple description video coding.

1. Introduction

The recent advances of the Internet and computing technologies have opened up new opportunities to multimedia applications, where video services such as Internet Protocol Television (IPTV), Internet TV, video sharing, and video podcast have gained significant popularity. Despite recent advances, video streaming over the Internet still presents many challenges. Peer-to-Peer (P2P) networks have emerged as a valuable infrastructure to support video streaming (e.g., videoconferencing applications), and have started replacing traditional content delivery networks.

A P2P communication infrastructure is formed by a group of nodes located in a physical network. These nodes build a network abstraction on top of the physical network known as an overlay network, which is independent of the underlying physical one. An important advantage of peer-to-peer systems is that all available resources are provided by the peers. In a P2P media delivery system, each peer can take the role of both a server and a client at the same time. As a client, a peer receives data from other peers, while as a server, a peer forwards data to other peers. In contrast to Internet Protocol (IP) multicast and content delivery networks, P2P based media delivery systems do not require a dedicated infrastructure. However, each extension of the P2P system architecture by the activation of a new peer implies the increase in the system demand and capacity.

P2P video streaming systems aim to maximize the throughput while video quality should be delivered in a scalable fashion to a set of requesting peers with different upload capacities. Video transmission in such systems is exposed to variable transmission conditions. A scheme that has been shown to provably maximize the overall throughput during a multicast session is Mutualcast (Li *et al.*, 2005). This scheme maximizes the throughput by exploiting the full upload capacity of all participating peers. Mutualcast is based on a fully-connected overlay mesh and the links between peers are established via TCP (Transport Control Protocol) connections. In a fully-connected overlay mesh, the number of TCP connections for n peer sites (including the source peer) is given by

$$\frac{n(n-1)}{2}. \qquad (1)$$

TCP handles flow control, congestion control and reliable data delivery, while the TCP buffers are used as redistribution queues. Although Mutualcast achieves the maximum possible multicast throughput in P2P networks with constrained upload capacities, it has some limitations derived from its TCP based communication mode. A drawback of using TCP is the potentially large delay values implied by its retransmission mechanism. Time-delay plays a very important role for modeling networks (Wang *et al.*, 2009; Morawski and Zajączkowski, 2010). Control systems or multimedia systems are examples of time-delay systems. For example, large delay values are completely unacceptable for real-time video applications. In contrast, the UDP (User Datagram Protocol) allows predictable and reduced delay, but it does not guarantee packet delivery and correctness.

In this paper, we address these issues and propose a solution for P2P video multicast based on Scalable Video Coding (SVC) and the user datagram protocol. Alternatively, a video multicast scheme based on Layered Multiple Description Video Coding (LMDVC) and the UDP is proposed. Our method is inspired by Mutualcast (Li *et al.*, 2005) and the multiple description video coding model presented by Essaili *et al.* (2007), Chou *et al.* (2003) as well as Puri and Ramchandran (1999). In much the same way as in Mutualcast, we defined the fully meshed topology to achieve the maximum possible throughput, but we establish the links between peers via UDP connections. We also propose a flow control mechanism, which is integrated within the application layer. Our flow control mechanism provides efficient adaptation to bandwidth variations of individual peers.

Scalable video coding can provide the encoding of a high quality video bit stream, which contains some subset bit streams that can themselves be decoded with a complexity and reconstruction quality similar to that achieved using the existing H.264/AVC (Advanced Video Coding) design with the same quantity of data as in the subset bitstream (Schwarz *et al.*, 2007). Using scalable video coding, parts of a video stream can be removed and the resulting substream constitutes another valid video stream for some target decoder. Thus, we distribute video of different quality to requesting peers with different bandwidth characteristics or when the network characteristics are time-varying. On the other hand, Multiple Description Coding (MDC) divides a single media stream into several sub bit streams. Reconstruction quality improves with the number of descriptions received in parallel (Goyal, 2001).

From LMDVC, we derived the idea of how to encode the video content. Using LMDVC, the scalable media stream is fragmented into multiple independent descriptions of equal importance, and reconstruction quality improves proportionally with respect to the number of descriptions received. LMDVC is targeted to best effort transmission, because it increases the robustness of the coded stream and reduces error propagation. Mainly, our work has been motivated by the aim to deliver video of high quality and low delay among a small number of participants (i.e., 5 to 10 peers) connected to the Internet without the involvement of a costly streaming infrastructure.

Our contribution can be then summarized as follows:

- We define a mesh-based P2P structure, which maximizes the overall throughput.
- A flow control mechanism is integrated at the source to (a) detect the varying network conditions and (b) effectively use the bandwidth between the source and each peer.
- A scalable video coding scheme is integrated into a meshed-P2P streaming system in order to deliver differentiated quality to peers with different capacities.
- Multiple description coding and scalable video coding are combined in order to alleviate the packet loss problem in a meshed-P2P streaming system.

This paper is organized as follows. We discuss related work in Section 2. In Section 3, our delivery model is introduced. Then, we use this model to describe our network infrastructure initialization, a scalable video coding scheme, combining SVC and MDC, and flow control scheme. In Section 4, we evaluate the performance of our model by means of experiments realized in PlanetLab (Peterson *et al.*, 2002). Section 6 concludes the paper.

In this paper we present an extended version of our other work (López-Fuentes, 2010). Specifically, the following new material has been added: An exposition about the features and benefits of scalable video coding for video streaming are introduced in Section 2. Two new subsections have been added in the delivery mechanism section. The first new subsection gives a brief analytical framework of Mutualcast based on the work of Li *et al.* (2005), while the second new subsection explains the network structure initialization of our delivery mechanism. Additionally, a scalable video coding pseudocode has been added in this section in order to give a formal approach of this technique in our work. Information about the JSVM software and specific pictures comparing the different reconstruction quality by using scalable video coding are introduced in Section 5.

2. Related work

Currently, video delivery is a very active working area in the research community and several video coding techniques have been investigated. In this context, scalable video coding and multiple description coding are well established and widely studied concepts. These techniques are often proposed in video streaming systems to gain robustness during a transmission.

MDC has emerged as a promising scheme to enhance the error resilience of a video delivery system (Wang *et al.*, 2005). It is a technique that generates two or more data streams containing descriptions of the source. Each description can be decoded independently of the others to provide baseline quality. However, the decoded videos from the different descriptions can be combined in order to provide a higher quality video. The descriptions can be individually packeted and sent from the source node to the receiver node through either the same or separate physical channels or paths. MDC can be available as Balanced Multiple Description Coding (BMDC) or Unbalance Multiple Description Coding (UMDC). In a balanced MDC scenario, all descriptions (two or more) are equally important and they mutually refine one another. In contrast, an unbalanced MDC scenario generates descriptions of different quality. In a scenario with two descriptors, one description is of quality while the second description of Low Quality (LQ). The LQ description is used to add redundancy and to conceal errors in the HQ description. The main difference between both the approaches is that in UMDC the protection stream is an LQ stream, while in BMDC the protection is provided by a Forward Error Correction (FEC) code.

Scalable video coding is a technique that encodes the video into layers. It incorporates the following scalability modes:

- Temporal scalability: the subset bitstream represents lower temporal resolution. With the subset bitstream, a part of frames in one GOP can be decoded.
- Spatial scalability: the lower subset bitstream can only playback a video with a lower frame size.
- SNR (Signal-to-Noise Ratio)/fidelity scalability: the base layer bitstream can only playback a video of very low quality, and the more enhanced layers the client receives, the better quality the video is.
- Combined scalability: it is a combination of all three or two modalities above.

For quality scalability, there are three types of scalability coding, i.e., Coarse Grain Scalability (CGS), Medium Grain Scalability (MGS) and Fine Grain Scalability (FGS). CGS coders divide a video stream into multiple layers, which provide limited rate scalability at the layer level. Coarse-grain SNR scalable coding is achieved using the concepts of spatial scalability. The same inter-layer prediction mechanisms are employed, but the up-sampling operations are omitted. The CGS only allows a few selected bit rates to be supported in a scalable bit stream. In general, the number of supported rate points is identical to the number of layers. Switching between different CGS layers can only be done at defined points in the bit stream. Furthermore, the CGS concept becomes less efficient when the relative rate difference between successive CGS layers gets smaller (Wien *et al.*, 2007).

SVC starts with the base layer, which contains the lowest level of the spatial, temporal and quality perspective detail. Additional layers called enhancement layers can increase the quality of the video stream. An enhancement layer is called a spatial enhancement layer when the spatial resolution changes with respect to its reference layer, and it is called a fidelity enhancement layer when the spatial resolution is identical to that of its reference layer (Schwarz and Wien, 2008). Scalable video coding introduces new video coding techniques which provide the following features: reduced bitrate, reduced spatial-temporal resolution and coding efficiency comparable with non-scalable video systems. SVC application areas include video surveillance systems, mobile streaming video, multichannel video distribution, and multi-party video conferencing. In this paper, we use the JSVM (Joint Scalable Video Model) software (Reichel *et al.*, 2007) as a codec to provide SNR scalable bitstreams.

The main difference between MDC and SVC is that, in MDC, video quality is improved with the number of descriptions received in parallel, while in SVC the enhancement layers are applied to improve stream quality. Several works about SVC and MDC are reported in the literature. Scalable video has been used to adapt the same video quality for different videos distributed from different sources to multiple requesting peers (López-Fuentes and Steinbach, 2008). A comparative study of MDC and SVC for a wide range of scenarios using network simulations is presented by Singh *et al.* (2000). In these scenarios, the base layer is transmitted via TCP, while the enhancement layers are transmitted via UDP. All descriptors MDC are transmitted via UDP. The performance of SVC and MDC using multiple paths is compared by Wang *et al.* (2002). Both previous studies conclude that MDC has advantages over SVC for applications with very stringent delay constraints or for networks with a long RTT (Round Trip Time). The error-resilience capabilities of MDC and SVC are studied by Lee *et al.* (2003) through extensive experiments. The results provide a most comprehensive performance comparison between these two techniques.

The performance of specific implementations of MDC and SVC over error-prone packet switched networks is examined by Chakareski *et al.* (2003). The authors compare the performance of both techniques using different transmission schemes. The authors conclude that the performance between MDC and SVC depends on the employed transmission scheme. On the other hand, in other works, SVC and MDC are combined in order to exploit the individual benefits of these schemes. For example, a combination of scalable video and erasure coding to generate multiple descriptions is introduced by Taal *et al.* (2004). In this work, the system estimates the bandwidth of the nodes to calculate an optimal allocation of the rates for all layers. A novel idea in this work is the application of erasure code to each layer separately, and then splitting up the resulting data into M descriptions. A mechanism to transform a scalable bit stream into a robust MDC packet stream is proposed by Puri and Ramchandran (1999). In this scheme, the layers are encoded applying a progressive protection related to their importance.

Layered multiple-description video coding to achieve robustness for unreliable channels and adaptability to heterogeneous clients is proposed by Chou *et al*. (2003). This scheme seems well suited for multicast scenarios where a heterogeneous client collection can be reached via multiple distribution trees. A multiple description coding algorithm, based on weighted signal combinations, is proposed by Zhao *et al*. (2007). This system uses a scalable codec in MDC in order to provide multiple descriptions and scalability at the same time. A multiple description scalable video coder based on the scalable video extension of H.264/AVC has been proposed by Zhao *et al*. (2009). The introduced method generates two descriptions with the same enhancement layers and the same motion vector.

The work we are presenting in this contribution owns some similarities to these previous works in the sense that we use SVC or, alternatively, LMDVC for content delivery. However, several features make our approach innovative. Our encoding schemes are designed for meshed P2P networks based on UDP, to which we additionally integrate a flow control scheme.

3. Delivery mechanism

In order to merge the advantage of both TCP and UDP protocols, we propose a communication structure, which integrates the protocols in a redistribution purpose. The general scenario of the proposed system is presented in Fig. 1. The communication structure contains one source node and four clients nodes (three requesting peers R_1, R_2, R_3, and one helper peer H) which are able to receive and forward the streams. UDP links are used to send the video blocks from the source to each requesting peer, while the TCP links are used by the source for receiving the feedback from the requesting peers. Similar to Mutualcast, each peer receives a single block from the source for redistribution. Thus, as shown in Fig. 1, the blocks X_1, X_2, and X_3, have been sent to the peers R_1, R_2 and R_3, respectively. The block X_4 is sent to the helper peer H for redistribution to the requesting peers R_1, R_2 and R_3. The helper peer H is not interested in receiving the video and just contributes its upload capacity during distribution. The strategies and incentives to select the helper peers are not considered in this paper. If the source site has abundant upload capacity, it sends one copy of the block X_5 directly to each requesting peer R_i. We assume in our approach that the upload capacity of the peers is constrained, while the download capacity is infinite and all participating peers are present during a streaming session.

The transmission control protocol is a connection-oriented protocol and provides a reliable data transfer with congestion and flow control (Postel, 1980b). Using TCP we do not need to consider the packet losses because of

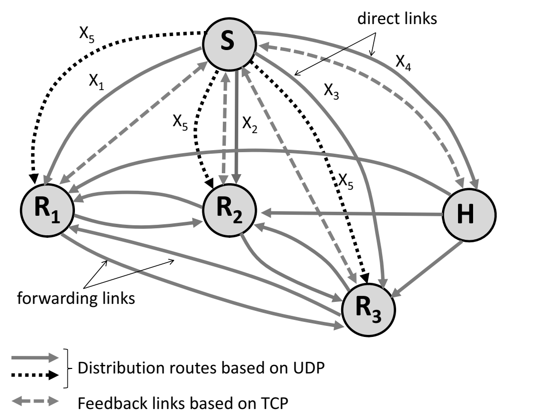

Fig. 1. Mesh-based P2P media streaming scheme with UDP and TCP links.

the guarantees that all data sent from one host to another will be received without duplication or losing data. On the other hand, the user datagram protocol is a very simple message-based connectionless protocol and does not guarantee delivery and duplicate protection of the datagrams (packets) sent (Postel, 1980a). Using UDP, the datagrams may be duplicated, dropped, delayed, or may arrive out of order. However, UDP has lightweight because of the capacity of the minimum header size of an IPv4 UDP header, which is only 8 bytes.

The distribution scheme shown in Fig. 1 contains four TCP connections, which are used in the initialization of the network structure and in the flow control of the UDP data transfer. Each peer has only one TCP connection to the source node. This distribution system is also built on ten UDP connections, four of which are direct data links and connect source node with each client node ($S - R_1$, $S - R_2$, $S - R_3$, and $S - H$). Three UDP connections forward the data link and connect each pair of the peer nodes ($R_1 - R_2$, $R_1 - R_3$, $R_2 - R_3$). Finally, three UDP connections are direct data links and connect helper node with each requesting peer ($H - R_1$, $H - R_2$, $H - R_3$). All direct data links are one-way communications, while the forwarding data links are two-way communications.

3.1. Analytical framework. A theoretical analysis of Mutualcast is given by Li *et al*. (2005). In this analysis, the authors prove that Mutualcast achieves the maximum possible throughput for peer-to-peer networks with a constrained upload bandwidth. Our analytical framework briefly reproduces this scenario. Similarly to Mutualcast (Li *et al*., 2005), the distribution scheme is composed of a source S of upload capacity C_S, N_1 requesting peers denoted as R_i with upload capacity C_{R_i}, and N_2 helper peers H_i with upload capacity C_{H_i}. The participating peers for this scenario scheme are shown in Fig. 2.

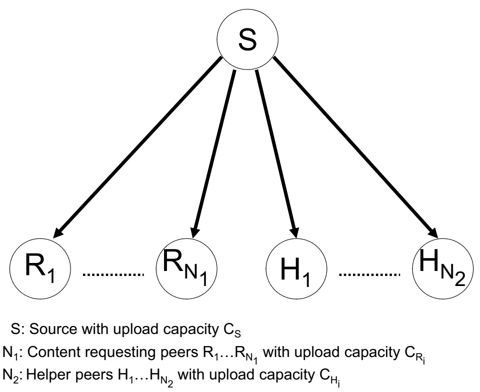

Fig. 2. Participating peers in the distribution scheme.

The upload capacity (bandwidth) of the source bounds the content distribution capacity in the system. The exhaustion of the source upload capacity is given by the following formula:

$$B_R + B_H + N_1 B_D = \sum_{i=1}^{N_1} B_{R_i} + \sum_{i=1}^{N_2} B_{H_i} + N_1 B_D. \quad (2)$$

The first component of the formula (2) represents the amount of data that are sent from the source to the N_1 content requesting peers for redistribution. The second component corresponds to the data being sent from the source to the N_2 helper peers and the last term represents the block of video data B_D sent directly to the N_1 content requesting peers from the source. The amount of data B_{R_i} sent from the source to each requesting R_i is limited by the upload capacity C_{R_i} and can be calculated from the following formula:

$$B_{R_i} = \frac{C_{R_i}}{N_1 - 1}. \quad (3)$$

It can be also interpreted in terms of the content, which has to be redistributed to the $N - 1$ other content requesting peers.

Similarly, for the helper peers, the amount of data B_{H_i} sent from the source to each helper peer H_i is limited by the upload capacity C_{H_i} and is given by

$$B_{H_i} = \frac{C_{H_i}}{N_1}. \quad (4)$$

Thus, B_R and B_H can be calculated using

$$B_R = \sum_{i=1}^{N_1} B_{R_i} = \sum_{i=1}^{N_1} \frac{C_{R_i}}{N_1 - 1} = \frac{N_1}{N_1 - 1}\bar{C}_R, \quad (5)$$

$$B_H = \sum_{i=1}^{N_2} B_{H_i} = \sum_{i=1}^{N_2} \frac{C_{H_i}}{N_1} = \frac{N_2}{N_1}\bar{C}_H, \quad (6)$$

where $\bar{C}_R$ and $\bar{C}_H$ are the mean upload capacities of the N_1 requesting peers and the N_2 helper peers, respectively.

The distribution throughput θ of the Mutualcast network is limited only by the upload capacity of the source, if the following condition is satisfied: $C_S \leq (B_R + B_H)$ (see also Li *et al.*, 2005). We can represent the overall distribution throughput θ in the following way:

$$\Theta = C_S. \quad (7)$$

It can be observed that the upload capacity of all the requesting peers R_i and helpers peers H_i might not be exhausted. The distribution throughput is defined as the amount of content sent to the requesting peers per second. For $C_S > (B_R + B_H)$, the distribution throughput θ is calculated using the following formula (Li *et al.*, 2005):

$$\Theta = B_R + B_H + \frac{(C_S - B_R - B_H)}{N_1}. \quad (8)$$

It can be observed that the upload capacity of all the requesting peers R_i and helper peers H_i is exhausted, but the source still has abundant upload capacity. Thus, the rest of the upload capacity of the source is divided among all the requesting peers R_i. For our scheme, we assume that the overall throughput is same as the Mutualcast throughput. This overall throughput considers the contents distributed via the UDP links and the feedback messages delivered via the TCP links.

3.2. Network structure initialization. We assume in network structure initialization that all peers know the address and listen-port of the source node. The requesting peers and the helper peer send their own information to the source in order for the source to forward this information to all participating peers. Our initialization strategy is shown in Fig. 3, and can be defined by the following procedure: Initially, the source node starts listening on a predefined port and waits for the TCP connections request. Then, the requesting and helper peers send their connection requests to the source node and wait for the acknowledgment from the source. We use the initialization packets for the distribution of the request and acknowledgment messages. These packets can be directly written to TCP sockets.

Once the acknowledgment message is received by the requesting peer or helper peer, the peers send information about their UDP listen-ports to the source node. The source node gets a list with the IP address and listen-port number of all requesting and helper peers if there is a sufficient number of active requesting peers in the system. This number of active peers is a parameter of the procedure. Using this list, the source node sends the IP address list and parameters, such as the frame rate, structure size and node order, to all peers. It is necessary for the identification by the requesting or helper peers of the addresses and UDP listen-ports of the other peers in the system.

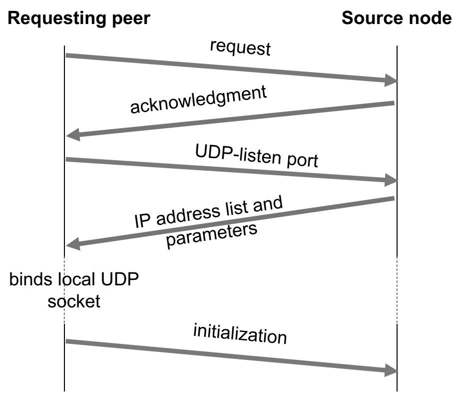

Fig. 3. Network structure initialization.

In the next step, the UDP connections are initialized. Each peer, including source node, binds a local UDP socket to every one of the others. Since UDP is connectionless, the UDP links are regarded as established. In the last step, each requesting peer or helper peer sends the signal "initialization finished" to the source node, which indicates that the structure of the network has been initialized and all links are ready for content distributing.

3.3. Scalable video coding. The main challenge in the delivery of video to a heterogeneous and dynamically varying network is to define the model that (a) maximizes the quality of the video received by the users, and (b) simultaneously manages to deal with the bit rate limitations (Chakareski *et al.*, 2003). Scalable video coding is a well-known method that can be successfully applied to solve the rate matching problem (Singh *et al.*, 2000). SVC can be classified as a layered video codec, and it is suitable for different use-cases. For example, in a multicast system with clients of different capabilities, the scalable bit stream allows delivering decodeable and presentable quality of the video depending on the device's capabilities. Here, the presentable quality criterion is expressed as the resolution, frame rate and bit rate of a decoded operation point of the scalable video stream.

SVC addresses the issue of reliably delivering video to diverse systems over heterogeneous networks extending the target applications of the H.264/AVC standard to enable video transmission with heterogeneous clients. To achieve it, SVC uses available system resources, in the case of the lack of a prior knowledge of the downstream client capabilities, resources, and variable network conditions (Schwarz and Wien, 2008). In this scenario, SVC enables robust transmission with graceful degradation in the presence of errors and bit stream adaptation. For example, clients may have different display resolutions, systems may have different intermediate storage resources, and networks may have varying bandwidths and loss rates.

SVC encodes the video into one Base Layer (BL) and one or more Enhancement Layers (ELs). The BL provides a basic level of quality, while the enhancement layers refine base layer quality. The base layer can be decoded independently of the enhancement layers. However, alone, the enhancement layers are not useful. Thus, the base layer represents the most critical part of scalable video representation. SVC provides flexibility, because when the network capacity is not sufficient, only a subset of the layers is distributed to the requesting peers and they can still display the video, though of reduced quality. When the network capacity increases, the requesting peers can receive more layers and thus reconstruction quality can be improved. Because peers with a higher bandwidth request more layers and achieve higher reconstruction quality than the requesting peers with a less bandwidth, SVC also offers differentiated video quality to peers with different bandwidth characteristics. In Fig. 4 we present the scalable video coding pseudocode used in this work. This general procedure must be adapted by the flow control scheme for each operation mode (SVC and LMDVC mode).

```
Procedure scalable video coding
begin
PSNR ← 0            /* PSNR is used as quality metric */
For each codification of video V_i make M diferent layers L_i
    if i=1, then L_i is base layer
        if i≠1 then L_i is an enhacement layer and
                different L_i+1, ..., L_i+M can be defined

For each reception of layer L_i at a node
    if L_i=1 is received then
        decode L_1 and calculate a basic PSNR for video V_i
        for all L_i where i=2,...,M
                accept and decode L_i
            for each L_i≠1 received
                increase PSNR of video V_i
            endfor
        endfor
    else all L_i≠1 are not useful
end
```

Fig. 4. Scalable video coding pseudocode.

To adapt the scalable video coding procedure to our distribution scheme (see Fig. 1), we assume that the base layer or enhancement layer can be represented by block X_i. The source node sends the BL to all requesting peers. However, notice that not all of them receive the EL. It depends on the upload capacity of each requesting peer. The BL is directly delivered by the source to each peer, because of the BL highest priority. The priorities of the enhancement layers can be various (e.g., EL_1 has a higher priority than EL_2). If the source has abundant upload capacity after the BL has been sent, then the source sends the EL according to the respective layer priority. If the upload capacity of all clients is large enough, it is used by the source by sending different ELs to each requesting peer for redistribution. Following this principle, one EL forwarded from another peer is considered with a higher priority than one EL sent directly from the source. In our approach, the source uses a swapping strategy to select a competent peer for each EL that has to be redistributed. However, if the upload capacity of a selected peer is insufficient to forward a copy of a given EL to all peers, the source sends it to the rest of the requesting peers. Peers with similar upload capacity receive from the source the same number of ELs for the redistribution. Peers with heterogeneous upload capacity receive different numbers of ELs. The network dynamics cause the dynamic changes in the transfer rules for enhancement layers application.

3.4. Combining SVC and MDC. Multiple description coding has been proposed as an effective tool for the elimination of the packet losses on the Internet (Goyal, 2001). Based on this idea, we combine MDC and SVC coding methods to solve the problem of layer dependencies in the meshed-P2P video multicast. In layered multiple description video coding, each peer receives a number of descriptions based on its upload capacity. In contrast to scalable video, each description can be independently decoded. In this case, peers with high upload capacity can receive more descriptions and consequently increase their reconstruction quality. The idea of LMDVC is presented in Fig. 5. In this scenario, the source encodes three layers (BL+2EL) into three descriptions. The ELs are partitioned into small segments of predefined size. The more important a layer is, the more protection it should receive. Thus, the BL is protected by two FEC blocks, while EL_1 is divided in two parts and protected by only one FEC block. EL_2 is not protected, but this layer is less important than the BL and EL_1. A peer can decode the BL when it receives one description. For EL_1, two descriptions are required. In this case, each requesting peer directly receives a single description from the source, while the rest of the descriptions are received from the other peers. Any peer that gets even a small amount of the EL automatically increases its PSNR (Peak Signal-to-Noise Ratio) due to the inherent difference.

3.5. Flow control scheme. The video delivery mechanism in our mesh-based P2P video multicast scheme is

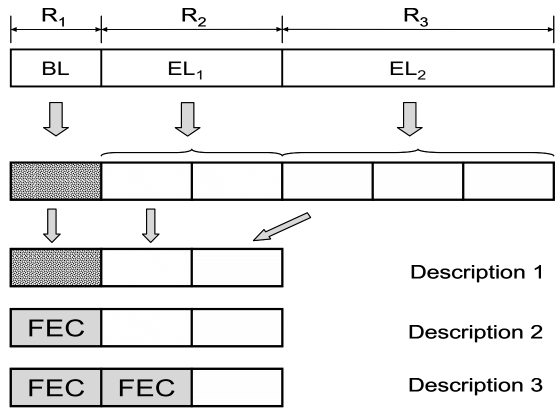

Fig. 5. Encoding scalable layers into multiple descriptors.

based on UDP connections. The delays in the system generated by the UDP connections are usually smaller than those generated by the TCP connections. The flow control in the system is needed for the adaptation of the data transfer rate to the temporal network capacity and for the reduction of the packet loss rate. In our system, central flow control is integrated with the source and works in two independent modes.

In the first mode, just scalable video coding is applied. The video is encoded at M different layers, L_1 corresponds to the base layer, and L_2 to L_M correspond to the enhancement layers. Each layer L_i has a fixed length equal to R_i, $i = 1, 2, \ldots, M$. Each layer L_i is partitioned into small segments of predefined size l, resulting in k_i segments. The packet length is an integer multiple of the segment l size. If L_i corresponds to base layer L_1, it is delivered to the peers R_i from the source directly. If L_i is different to L_1, then L_i is an EL, and thus the source selects the peer R_i with enough upload capacity C_i to redistribute this EL. During the ELs distribution, the requesting peers R_i can receive packets $P^{L_i}_{R_{i,j}}$ from the source via direct links $R_{s,i}$ or packets $Q^{L_i}_{R_{i,j}}$ from other peers via forward links $R_{i,j}$. The source S registers the number of packets $P^{L_i}_{R_{i,j}}$ and $Q^{L_i}_{R_{i,j}}$ that has been delivered to each peer R_i via either the direct links or forwarding links. Using a report packet, each requesting peer R_i returns an acknowledgement about $P^{L_i}_{R_{i,j}}$ and $Q^{L_i}_{R_{i,j}}$. The reported interval is given in frames. Thus, the source sends the frame rate in the initialization packet and each peer R_i computes the number of frames F_i to be received during a given time. In order to adapt the data transfer rate to the current network situation, the requesting peers must send an acknowledgement to the source as soon as possible. The peer report depends on some statistical measures: when the first packet P_1 of the second frame F_2 interval arrives to each peer, it sends a report packet to the source specifying the previous report delay. The source S starts the acknowledgement procedure when the report packet is received. The report packet also includes the sequence number s_i, while the link number $R_{i,j}$ is respectively used to find the counter of the corresponding link and s_i to found the counter with the previous number of packets sent via link $R_{i,j}$. Then, the source S computes the packet loss rate for all links, and if necessary reorganizes the content distribution.

In the second mode, the LMDVC method is used, and we assume that the descriptions produced by LMDVC have the same size and all requesting peers R_i receive the same amount of packets. Then, the source S does not need to know the network conditions for every single link, and a simplified version of the acknowledgement mechanism can be used, because the source S only needs one counter for the packets $P^{L_i}_{R_{i,j}}$ sent to each single direct link.

Like SVC, the flow control mechanism is started by the source after it receives the report packet from the peers. However, the flow control mechanism in LMDVC works differently, because it only modifies the FEC (Forward Error Correction) of each layer. In other words, the flow control mechanism only reduces or increases redundancy to protect the BL against a packet loss. The flow control mechanism aims at maximizing the reconstructed video quality in each requesting peer in both modes. Our mechanism also redistributes the packet distribution load between the links in order to avoid generating significant delay among them.

4. Performance evaluation

In this section, we present a simple experimental evaluation of our model. A mesh-based P2P multicast prototype based on SVC and LMDVC has been implemented and tested on the PlanetLab infrastructure. PlanetLab is a globally distributed platform for testing and deploying new network services. The main purpose of using PlanetLab is to define a testbed for overlay networks (Peterson *et al.*, 2002). Researchers can use the Planetlab infrastructure to experiment with a variety of planetary-scale services, such as file sharing and network-embedded storage, content distribution networks, or multicast overlays. Therefore, we consider that PlanetLab constitutes a well-suited testbed to evaluate our proposed mechanism. Our implementation runs on Linux and consists of various programs written in C/C++. Each requesting peer runs a receiver module, which has been enabled with a sender/receiver mode. The feedback links are established via TCP connections, while the delivery links are established using UDP connections. Figure 6 presents a comparison of the throughput of a UDP link and a TCP link on the Internet. For this purpose, a 1-MB data-file was sent repeatedly (ten times) between two Planetlab nodes, within a time interval lasting several days. Figure 6 shows the results of the average measurements. With the results of these experiments it can be observed (see Fig. 6) that the UDP link performs better than the TCP link. Even though these experiments could be regarded as unsophisticated, an exhaustive comparison of the performance of these two protocols (UDP and TCP) is reported by Zhang and Schulzrinne (2004), who shows that UDP outperforms TCP in terms of delays, in congestion-situations where, as is known, TCP resends any packet that is lost, causing, with this, important delays.

Based on the detailed UDP effectiveness analysis (see Zhang and Schulzrinne, 2004), we use UDP links to distribute the video sequences from the source node to each requesting peer. Desired characteristics such as reliable data delivery, flow-control and the handling of redundancy and layers are automatically provided by the flow control mechanism. All the peers (except the source) send and receive packets at the same time. The distribution of blocks among the requesting peers is implemented using threads.

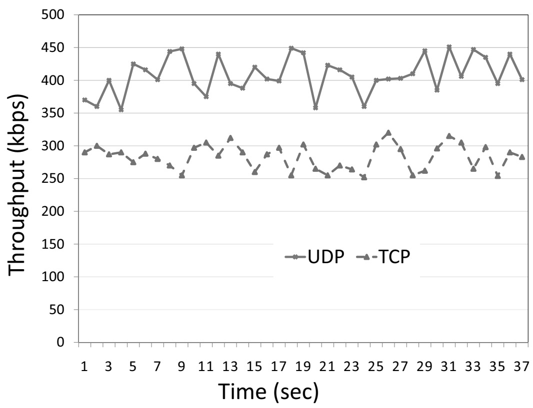

Fig. 6. Comparison of the UDP and TCP throughput on the Internet.

We select the BUS sequence (source file downloaded from EvalVid, 2010) for a CIF (Common International Format) size as our test sequence. A picture of the BUS YUV sequence is depicted in Fig. 7 (EvalVid, 2010). It is encoded using the JSVM software (JVT, 2005). JSVM is a scalable video coding codec used to encode and decode a video. JSVM can provide a bitstream that contains one or more subset bitstreams, which are derived by dropping packets from a larger bitstream. A base layer is a subset bitstream, and can playback a video with very low frame per second (fps), small size of resolution or low quality (PSNR). One or more enhanced layers can be encoded in order to obtain refinement. Our implementation uses the joint scalable video model as a codec to provide SNR scalable bit-streams. We use Coarse Grain Scalability (CGS) in this work. The JSVM software is the reference software for the scalable video coding project of the Joint Video Team (JVT) of the ISO/IEC MPEG (International Organization for Standardization/International Electrotechnical Commission—Moving Pictures Experts Group) and the Video Coding Experts Group (VCEG) of the ITU's Telecommunication Standardization Sector. JSVC is still under development and changes frequently. The reference software for the JSVM can be found on the Internet (Reichel *et al.*, 2007). The SNR scalable bitstreams are used as source video streams. However, the error concealment is by far not available for all configurations.

Fig. 7. Picture of the BUS YUV sequence.

We use the peak-signal-to-noise-ratio as the quality metric. For our experiments, we encode 60 frames with one BL and three ELs. The BL is encoded at 631.62 kbps, while EL_1, EL_2 and EL_3 are encoded at 864.58 kbps, 1165.24 kbps and 1594.42 kbps, respectively (Song, 2008). Using these rates, the BL, EL_1, EL_2 and EL_3 achieve video quality (PSNR) of 31.13 dB, 32.07 dB, 32.97 dB and 34.04 dB, respectively. The setting for this scenario is presented in Table 1.

Table 1. Example of bit-rates and PSNR values for the BUS video sequence. CIF size (352x288).

Bit-rate	Y-PSNR	U-PSNR	V-PSNR
631.620	31.1372	38.5909	39.9871
864.580	32.0703	39.3657	40.9710
1165.240	32.9767	39.8229	41.4464
1594.424	34.0488	40.7592	42.3460

Reconstruction quality for the BUS sequence is shown in Fig. 8 (Song, 2008). In Fig. 8(a), the base layer bitstream is reconstructed only. The picture shows that reconstruction quality is only acceptable. In Fig. 8(b), the picture is reconstructed from the base layer and three enhanced layers. Using refinement, we can see that reconstruction quality increases. However, all layers have identical picture spatial resolution. This scenario provides enough flexibility when the network capacity is not enough to deliver a video of high reconstruction quality. Then, the source can only distribute the base layer to the client, and it can still represent the video fluently. When the network capacity is better, the client can receive an additional layer, and base layer reconstruction quality can be improved.

For our LMDVC implementation, we use the bitstream produced by the JSVM. Thus, our LMDVC implementation is coding over scalable video coding. The implementation follows the steps: the source encodes three layers (BL+2EL) into three descriptions. A peer can decode the BL when it receives one description, while for EL1 two descriptions are required. Each requesting peer directly receives a single description from the source, while the rest of the descriptions are received from the other peers. Then, we evaluate the performance of our SVC approach and compare it with the LMDVC approach in terms of the throughput and packet loss.

For our experiments on PlanetLab, we select a small multicast group composed of four PlanetLab nodes. The source is located at Harvard University (`righthand.eecs.harvard.edu`), while the requesting peers are respectively located at the University of California, Santa Barbara (`planet1.cs.ucsb.edu`), the University of Kansas (`kupl1.ittc.ku.edu`), and the University of Oregon (`planetlab3.cs.uregon.edu`).

The first experiment (López-Fuentes, 2010) evaluates the adaptability of our flow control scheme to the bandwidth variations using scalable video (see Fig. 9). We can observe the forward link between R_1 and R_2 has a Packet Loss Rate (PLR) and cannot redistribute EL_1 from peer R_1 to peer R_2 during a time interval. Then, the flow control mechanism immediately adjusts the transfer plan and the source sends EL_1 directly to peer R_2. Thus, the throughput on the direct link is increased (around the 13th second), while the throughput of the forward link is reduced. When the forward link is again available (around the 15th second), the transfer plan is readjusted; we can observe that forward link throughput is increased, while the direct link throughput is reduced. We assume that peer R_2 achieves better video quality by receiving EL_1 from the source than if it is not received from peer R_1. In this experiment, we evaluate the performance of our system from a global outlook. Thus, the effect of change perceived by the receiver when a layer of video starts to be sent from a different peer during the transition period is not studied in this work. However, this effect can be an important topic to be considered in the future.

The second experiment (Song, 2008; López-Fuentes, 2010) evaluates our flow control mechanism using the LMDVC approach. In this case, the flow control mechanism only adds redundancy (forward error correction) (Fig. 10). Initially, no redundancy is assigned to BL and EL_1. After the 6-th second, the throughput goes down and the slice Unit Loss Rate (ULR) is increased in proportion to the packet loss rate. Then, the source adds redundancy

(a) (b)

Fig. 8. Comparison of different reconstruction quality for the BUS sequence: base layer only (a), base layer and three enhancement layers (b).

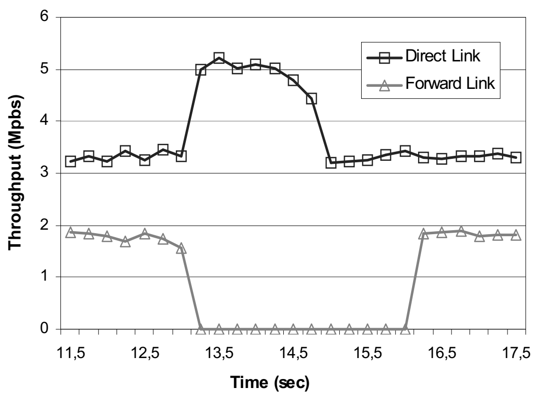

Fig. 9. Link performance during load redistribution realized by the flow control mechanism using SVC.

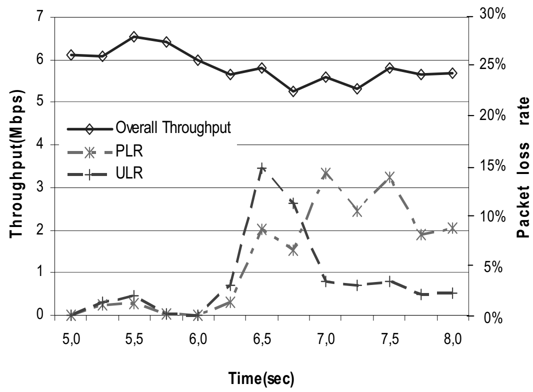

Fig. 10. System performance in the presence of packet loss and during redundancy addition by the flow control mechanism.

to the BL following the approach shown in Fig. 2. After this, the PLR remains high, but the ULR has been reduced to a lower level.

5. Conclusions and future work

In this contribution, we presented and evaluated scalable video coding and multiple description video coding strategies for video delivery to a small number of participating peers using meshed P2P networks.

Our solution streams the video via UDP connections, while feedback about the received rate is transmitted via TCP. A flow control mechanism is integrated into the source flow to offer differentiated video quality to peers with a different bandwidth. Scalable video coding is combined with multiple description coding to achieve scalable and robust transmission over unreliable networks. We evaluated our scheme on PlanetLab. The results show that our scheme achieves a promising performance to control the link flows. It allocates different number of layers to peers with different capacities or different levels of redundancy to the base layer. Using SVC, we can not only ensure subscribers that we will provide the minimum QoS, but we can also increase this level substantially depending on the upload capacity of the source. Scalable video coding, multiple description coding and forward error coding add overhead, which can have negative impact on the overall performance. The effect of this overhead over the overall throughput of our system is not quantified in these experiments.

The work proposed in this paper can be extended in different directions. For example, we use Coarse Grain Scalability (CSG) as input in our implementation. CSG provides coarse PSNRs and bit rates for choosing according to network conditions. However, in the future, Medium Grain Scalability (MGS) or Fine Grain Scalability (FGS) can be used in order to provide more enhanced layers for finer SNR scalability. Fine-grained scalability can also adapt most closely to the available network bandwidth. A mechanism to control the ping-pong effect can be integrated in the delivery mechanism. The goal of this mechanism will be to reduce the effect generated by the mobility of the rejected layers from other senders. On the other hand, a large number of multimedia applications require to protect the contents to be distributed over the Internet. Digital right management issues could be integrated in this proposed scheme as future work.

Acknowledgment

The author wishes to thank Xiao Song for his help programming a part of our mechanism used in our experiments. A part of the experiments in PlanetLab was carried out at the Institute of Communication Networks (LKN) at the Technical University of Munich, Germany.

References

Chakareski, J., Han, S. and Girod, B. (2003). Layered coding vs. multiple descriptions for video streaming over multiple paths, *Proceedings of the 11th ACM International Conference on Multimedia, MM 2003, Berkeley, CA, USA*, pp. 422–431.

Chou, P.A., Wang, H.J. and Padmanabhan, V.N. (2003). Layered multiple description coding, *Proceedings of the 13th International Packet Video Workshop, 2003, Nantes, France*.

Essaili, A.EL., Khan, S., Kellerer, W. and Steinbach, E. (2007). Multiple description video transcoding, *Proceedings of the International Conference on Image Processing, ICIP 2007, San Antonio, TX, USA*, pp. 77-80.

EvalVid (2010). Website of the Telecommunication Network Group, Technical University of Berlin, Berlin, http://www.tkn.tu-berlin.de/.

Fumagallia, M., Lancinia, R. and Tubaro S. (2006). A sequence-based error concealment algorithm for an unbalanced multiple description video coding system, *Signal Processing: Image Communication* **21**(10): 829–849.

Goyal V. K. (2001). Multiple description coding: Compression meets the networks, *IEEE Signal Processing Magazine* **18**(5): 74–93.

International Telecomunication Union (2005). Advanced Video Coding for Generic Audiovisual Service, *ITU-T Recomendation H.264 & ISO/IEC 14496-10 AVC v3*.

Joint Video Team (JVT) (2008). JSVM software manual, version 9.12.2, Heinrich Hertz Institute, Berlin, http://ip.hhi.de/imagecom_G1/savce/downloads/SVCReferenceSoftware.htm.

Lee, Y.-C., Kim J., Altunbasak, Y. and Mersereau, R.M. (2003). Performance comparisons of layered and multiple description coded video streaming over error-prone networks, *Proceedings of the International Conference on Communications, ICC 2003,Anchorage, AK, USA*, pp. 35–39.

Li, J., Chou, P.A. and Zhang, C. (2005). Mutualcast: An efficient mechanism for one-to-many content distribution, *Proceedings of the ACM SIGCOMM ASIA Workshop, Beijing, China*.

López-Fuentes, F.A. and Steinbach, E. (2008). Adaptive multi-source video multicast, *Proceedings of the IEEE International Conference on Multimedia & Expo, ICME 2008, Hannover, Germany*, pp. 457–460.

López-Fuentes, F.A. (2010). Adaptive mechanism for P2P video streaming using SVC and MDC, *Proceedings of the International Conference on Complex, Intelligent, and Software Intensive Systems, CISIS 2010, Cracow, Poland*, pp. 457–462.

Morawski, M. and Zajączkowski, A.M. (2010). Approach to the design of robust networked control systems, *International Journal of Applied Mathematics and Computer Science* **20**(4): 689–698, DOI: 10.2478/v10006-010-0052-0.

Peterson, L., Anderson, T., Culler, D. and Roscoe, T. (2002). A blueprint for introducing disruptive technology into the Internet, *Proceedings of the First Workshop on Hot Topics in Networks, HotNets-I 2002, Princeton, NJ, USA*, pp. 59–64.

Postel, J. (1980a). User Datagram Protocol, RFC 768, USC/Information Science Institute, Marina del Rey, CA.

Postel, J. (1980b). Transmission Control Protocol, RFC 793, USC/Information Science Institute, Marina del Rey, CA .

Puri, R. and Ramchandran, K. (1999). Multiple description source coding using forward error correction codes, *Proceedings of the Asilomar Conference on Signals, Systems, and Computers, Asimolar, CA, USA*, pp. 342–346.

Reichel, J., Schwarz, H. and Wien, M. (2007). Joint Scalable Video Model JSVM-11, *Doc. JVT-X202*, ISO/IEC MPEG&ITU-T VCEG, Geneva.

Schwarz, H., Marpe, D. and Wiegand, T. (2007). Overview of the scalable video coding extension of the H.264/AVC standard, *IEEE Transactions on Circuits and Systems for Video Technology* **17**(9): 1103–1120.

Schwarz, H. and Wien, M. (2007). The scalable video coding extension of the H.264/AVC standard, *IEEE Signal Processing Magazine* **25**(2): 135–141.

Singh, R., Ortega, A., Perret, L., and Jiang, W. (2000). Comparison of multiple description coding and layered coding based on network simulation, *Proceedings of the Visual Communications and Image Processing, SPIE 2000, San Jose, CA, USA*, pp. 929–939.

Song, X. (2008). *Comparing Multiple Description and Scalable Video in Mutualcast-based P2P Video Delivery*, Master thesis, Technical University of Munich, Germany, Munich.

Taal, J.R., Pouwelse, J.A. and Lagendijk, R.L. (2000). Scalable multiple description coding for video distribution in P2P networks, *Proceedings of the Picture Coding Symposium, PCS 2004, San Francisco, CA, USA*.

Wang, S.Y., Panwar, S., Lin, S. and Mao, S. (2002). Wireless video transport using path diversity: Multiple description vs. layered coding, *Proceedings of the International Conference on Image Processing, ICIP 2002, Rochester, NY, USA*, pp. 21–24.

Wang, S.Y., Reibamn, S. and Lin, S. (2002). Multiple description coding for video delivery, *Proceedings of the IEEE* **93**(1): 57–70.

Wang, D., Wu, S. and Okubo, S. (2009). Design of the state predictive model following control system with time-delay, *International Journal of Applied Mathematics and Computer Science* **19**(2): 247–254, DOI: 10.2478/v10006-009-0020-8.

Wiegand, T., Sullivan, G., Bjontegaard, G. and Luthra, A. (2002). Overview of the H.264/AVC video coding standard, *IEEE Transactions on Circuits and Systems for Video Technology* **13**(7): 560–576.

Wien, M., Schwarz, H. and Oelbaum, T. (2007). Performance analysis of SVC, *IEEE Transactions on Circuits and Systems for Video Technology* **17**(9): 1194–1203.

Zhang, X. and Schulzrinne, H. (2004). Voice over TCP and UDP, *Technical Report CUCS-033-04*, Computer Science Department, Columbia University, New York, NY.

Zhao, Z., Osterman, J. and Chen, H. (2009). Multiple description scalable coding for video streaming, *Proceedings of the 10th Workshop on Image Analysis for Multimedia Interactive Services, WIAMIS 2009, London, UK*, pp. 21–24.

Zhao, A., Wang, H., Cui, H. and Tang, K. (2002). Efficient multiple description scalable video coding scheme based on weighted signal combinations, *Tsinghua Science & Technology* **12**(1): 86–90.

Francisco de Asís López-Fuentes received a B.Sc. degree in electrical engineering from the Oaxaca Institute of Technology, Oaxaca, Mexico, in 1988. He received an M.Sc. degree in computer science minoring in networking from the Monterrey Institute of Technology (ITESM), Atizapán, Mexico, in 1998. From 2003 to 2008, he was a member of the research staff of the Media Technology Group in the Institute for Communication Networks at the Technical University of Munich, Germany, where he received the engineering doctorate in 2009. In December 2008, Dr. López-Fuentes joined the Department of Information Technology of the Autonomous Metropolitan University, Cuajimalpa, Mexico City, Mexico, where he is currently a professor-researcher for networks and distributed systems. From 1996 to 2003, he was a professor-researcher at the Institute of Electronic and Computer Science of the Technological University of Mixteca, Huajuapan, Mexico. From 1988 to 1994, he was a design engineer at the Engineering and Software Development Department of Siemens, Mexico. Dr. López-Fuentes has also served as a reviewer for a number of journals and conferences. His current research interests are in the area of networked and interactive multimedia systems, peer-to-peer networks, distributed systems as well as network security.

Received: 29 June 2010
Revised: 6 December 2010
Re-revised: 25 January 2011